

INICIAR SESIÓN

NUESTROS PLANES

TODOS LOS  
CURSOS

FORMACIONES

CURSOS

PARA  
EMPRESAS

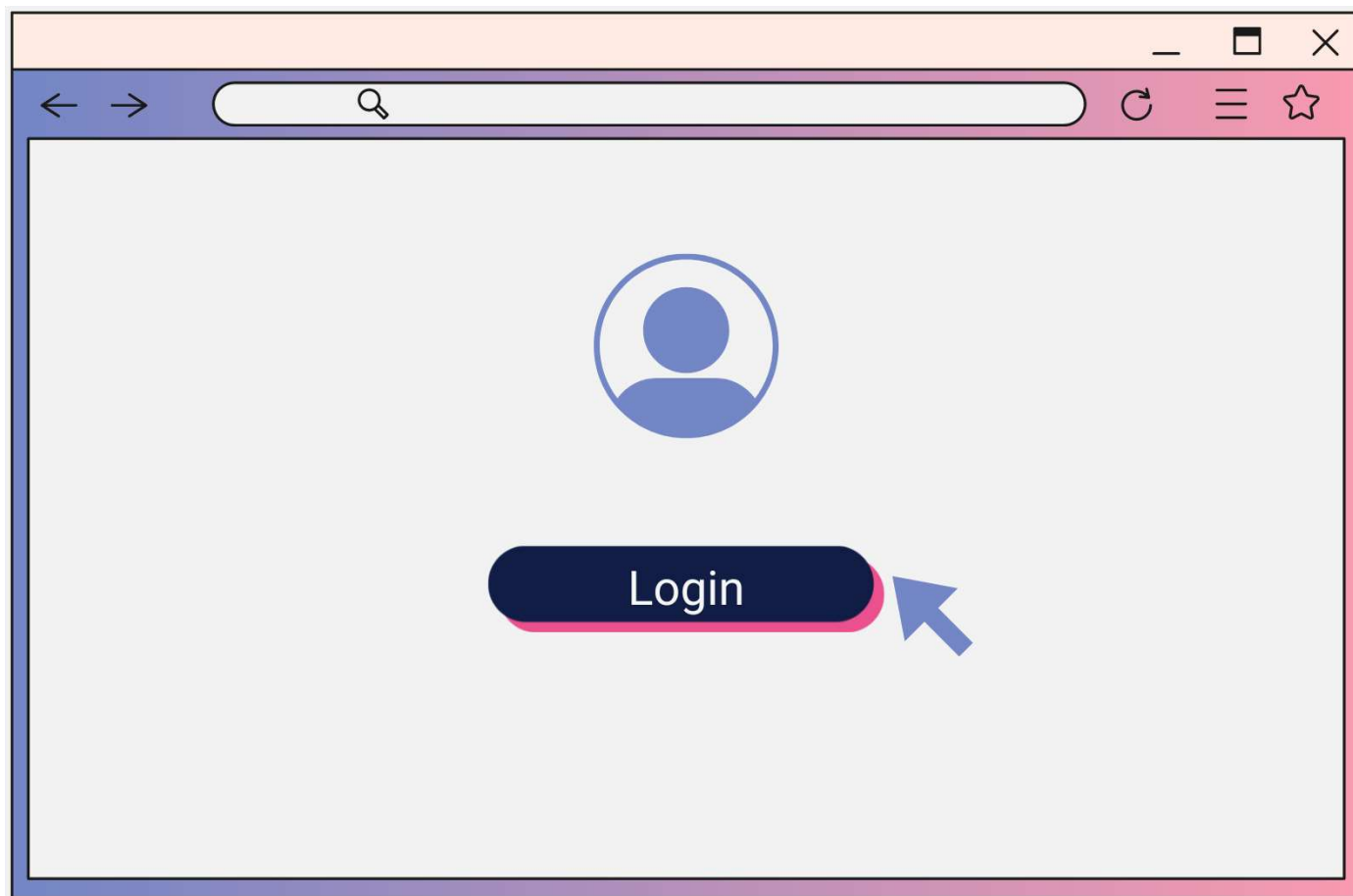
ARTÍCULOS DE TECNOLOGÍA

# Interfaces Gráficas con Eclipse WindowBuilder



Ellen Pimentel

22/08/2022



¿Es posible crear interfaces de usuario de forma rápida y sencilla en Java? Sí, con Java todo es posible y existe un aliado para esta tarea: el *plug-in* [WindowBuilder](https://www.aluracursos.com/blog/interfaces-graficas-con-eclipse-windowbuilder) para Eclipse.

Actualmente, la demanda de aplicaciones de escritorio es menor en comparación con otras plataformas, pero eso no significa que dejarán de existir, de hecho están las populares *Discord*, *Slack* y *GitHub Desktop*, desarrolladas en otros lenguajes. En **Java**, los ejemplos más conocidos son *Acrobat Reader*, *ThinkFree* y *Media Player*.

Si necesitas crear un proyecto personal, de estudio o incluso un sistema para una empresa con construcción de *GUI* (*Graphical User Interface* - Interfaz Gráfica de Usuario) utilizando Java, **WindowBuilder** es una excelente opción. Además de sus características, veremos un poco de su historia, estructura, método de instalación y componentes.

## El Plug-in WindowBuilder

Eclipse es un *IDE* (*Integrated Development Environment* - Ambiente de Desarrollo Integrado) con una estructura extensible y que permite agregar nuevas funciones y personalizar las existentes a través de complementos, así como eliminarlas. Se dividen en dos categorías: *plug-ins* de *UI*, que contienen aspectos de una interfaz de usuario o se basan en otros *plug-ins*, y *plug-ins* del *Core* que no son de interfaz y son independientes.

Con diseño *WYSIWYG* (*What You See Is What You Get* - Lo Que Ves Es Lo Que Obtienes), *WindowBuilder* es un *plug-in* que permite crear diseños inteligentes y sencillos usando el *drag-and-drop* (arrastrar y soltar) y ver el resultado en tiempo real, también podemos agregar controladores de eventos a los componentes y cambiar varias propiedades del *widget* a través del editor de propiedades.

La herramienta es bidireccional, es decir, puedes editar tanto el código fuente como cambiar los componentes en la ventana de diseño para realizar cambios en el *layout*. Ambas representaciones están sincronizadas, cualquier actualización se refleja en la otra.

La aplicación en desarrollo se puede adaptar fácilmente a múltiples idiomas y regiones sin cambios en su estructura interna, proveyendo recursos de forma automática. La herramienta también ofrece capacidades de estructuración inversa, es posible utilizar el código existente como punto de partida para un nuevo proyecto.

Otra característica importante está relacionada con los cambios mínimos en el código, la herramienta utiliza un estilo de microedición que garantiza realizar el cambio más pequeño posible en la fuente, conservando todo el código y el formato del usuario.

Además, *WB* (*WindowBuilder*) admite la herencia visual, creando una jerarquía de componentes visuales en una ventana, y también ofrece herramientas de transformación,

convirtiendo un tipo de componente en otro.

## Historia

Sus inicios anteceden a la creación del lenguaje **Java**, y a lo largo de su historia ha contado con la participación de grandes empresas:

**1991** – Lanzamiento original de *Smalltalk/V* por Cooper & Peters

*Smalltalk* es un lenguaje creado por Xerox y que popularizó el paradigma Orientado a Objetos. *Smalltalk/V* empezó a ofrecer un ambiente de ventana y fue desarrollado por la empresa Digitalk.

**1993** – Lanzamiento de *VisualSmalltalk* por ObjectShare

*Smalltalk/V* de Digitalk se convirtió en *VisualSmalltalk Enterprise*. Los cofundadores de Instantiations fundaron ObjectShare Systems, un proveedor líder de productos de *Smalltalk* que también desarrolló **WindowBuilder Pro**.

**1994** – Lanzamiento de *VisualAge Smalltalk* por ObjectShare

IBM utilizó las tecnologías *Smalltalk* integradas a la empresa *OTI (Object Technology International)* como base para *VisualAge*.

**1997** – Lanzamiento *VisualAge Smalltalk* por Instantiations

La Instantiations se hizo cargo del soporte de *VisualAge Smalltalk* de IBM.

**2003** – Nueva versión de Eclipse/Java para *SWT/RCP (SWT Designer)*

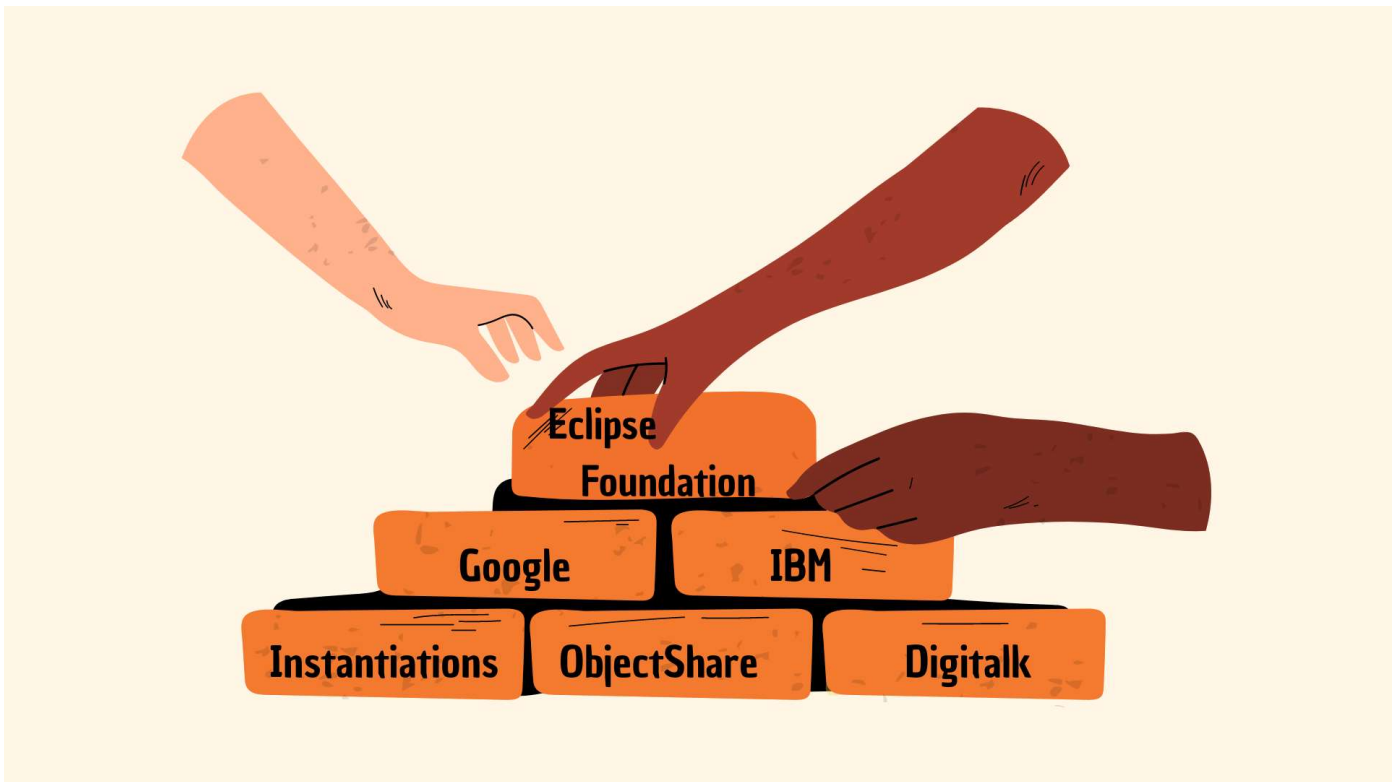
**2004** – Soporte *Swing* agregado (*Swing Designer*)

**2006** – Se agregó compatibilidad con *Google Web Toolkit (GWT Designer)*

**2009** – *Eclipse Community* premiada por el mejor Complemento Comercial

**2010** – Adquirido por Google y lanzado gratis al mundo

Solo los productos Java de Instantiations se volvieron propiedad de Google.



**WindowBuilder** se convirtió en un proyecto de código abierto y actualmente lo mantiene **Eclipse Foundation**, una fundación sin fines de lucro con miembros que apoyan y desarrollan **software** libre a través de **Eclipse Projects**.

## Estructura y composición

Su estructura se basa en bibliotecas populares como *Swing* y *SWT*, y también se compone por *GWT*, *RCP* y *XWT UI frameworks*. Además, WindowBuilder crea un árbol de sintaxis abstracta (*AST*) proporcionando acceso al código fuente y, a través del *GEF*, realiza la visualización y gestión de la presentación visual.

Estas siglas que lo componen y lo hacen potente y flexible, ¿qué significan? Para una mejor comprensión, aquí está la descripción de cada una:

**AST (Abstract Syntax Tree - Árbol de Sintaxis Abstracta)** - El código desarrollado se convierte en varios *tokens* y luego de un análisis sintáctico, la lista de *tokens* se transforma en un árbol que representa la estructura real del código. Cada nodo del árbol indica una construcción que tiene lugar en el código fuente. No existe un estándar *AST*, puede variar según el idioma y la herramienta utilizadas.

**GEF (Graphical Editing Framework - Estructura de Edición Gráfica)** - Ofrece un conjunto de herramientas para crear elementos de interfaz gráfica de usuario interactivos y

dinámicos, con la capacidad de agregar, mover, cambiar el tamaño y eliminar los elementos gráficos que los representan, lo que facilita el desarrollo de editores gráficos enriquecidos.

**GWT (Google Web Toolkit - Kit de Herramientas Web de Google)** - Es un conjunto de herramientas de código abierto con un marco de desarrollo de software Java y facilita la creación de aplicaciones *AJAX*. A través del compilador *GWT* es posible convertir la aplicación Java a JavaScript y HTML.

**RCP (Rich Client Platform - Plataforma de Cliente Enriquecida)** - Es un programa informático para crear aplicaciones Java basadas en la arquitectura Eclipse. Permite a los desarrolladores personalizar componentes y hacer que las aplicaciones sean portátiles y reutilizables. Por lo tanto, el conjunto mínimo de complementos necesarios para crear una aplicación se conoce como *Rich Client Platform*.

**SWING** - No es una abreviatura o sigla, sino el nombre de una de las librerías para desarrollar aplicaciones de escritorio en Java. Fue creado para reemplazar *AWT (Abstract Window Toolkit)*, la primera librería *GUI* del lenguaje, ya que sus componentes se implementaban de acuerdo a la estética de cada sistema operativo. La contribución de *Swing* fue ejecutar una sola interfaz en cualquier sistema operativo en el que se ejecutara.

**SWT (Standard Widget Toolkit - Kit de Herramientas de Widgets Estándar)** Es una biblioteca creada por IBM que proporciona widgets de interfaz de usuario integrados en el sistema operativo, pero la *API* es independiente del sistema operativo. Se puede utilizar para presentar una interfaz nativa a los usuarios.

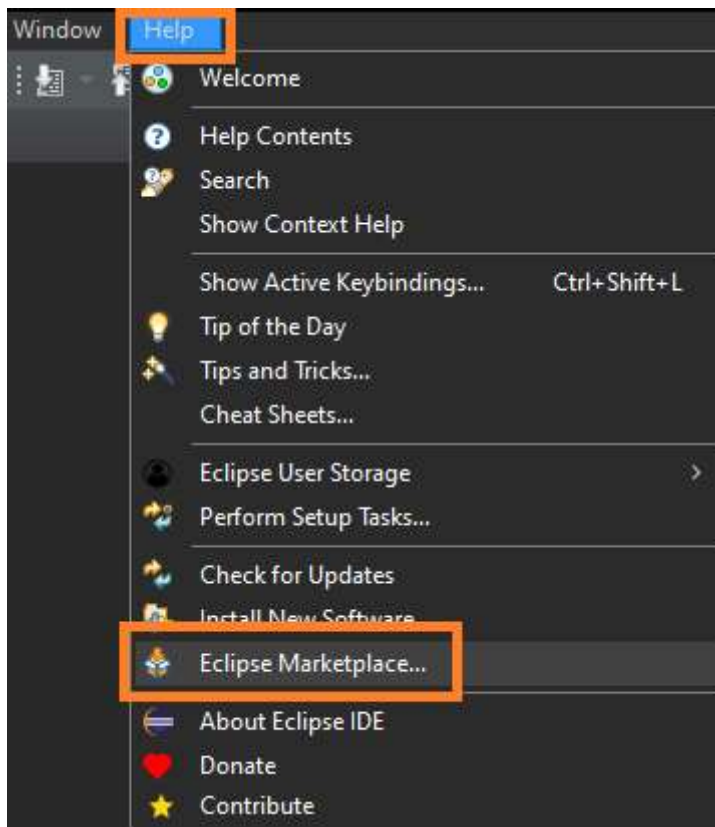
**XWT (XML Window Toolkit - Kit de Herramientas de Ventana XML)** Basado en el lenguaje de marcado XML, es un marco de interfaz de usuario declarativo potente y ligero diseñado para Eclipse.



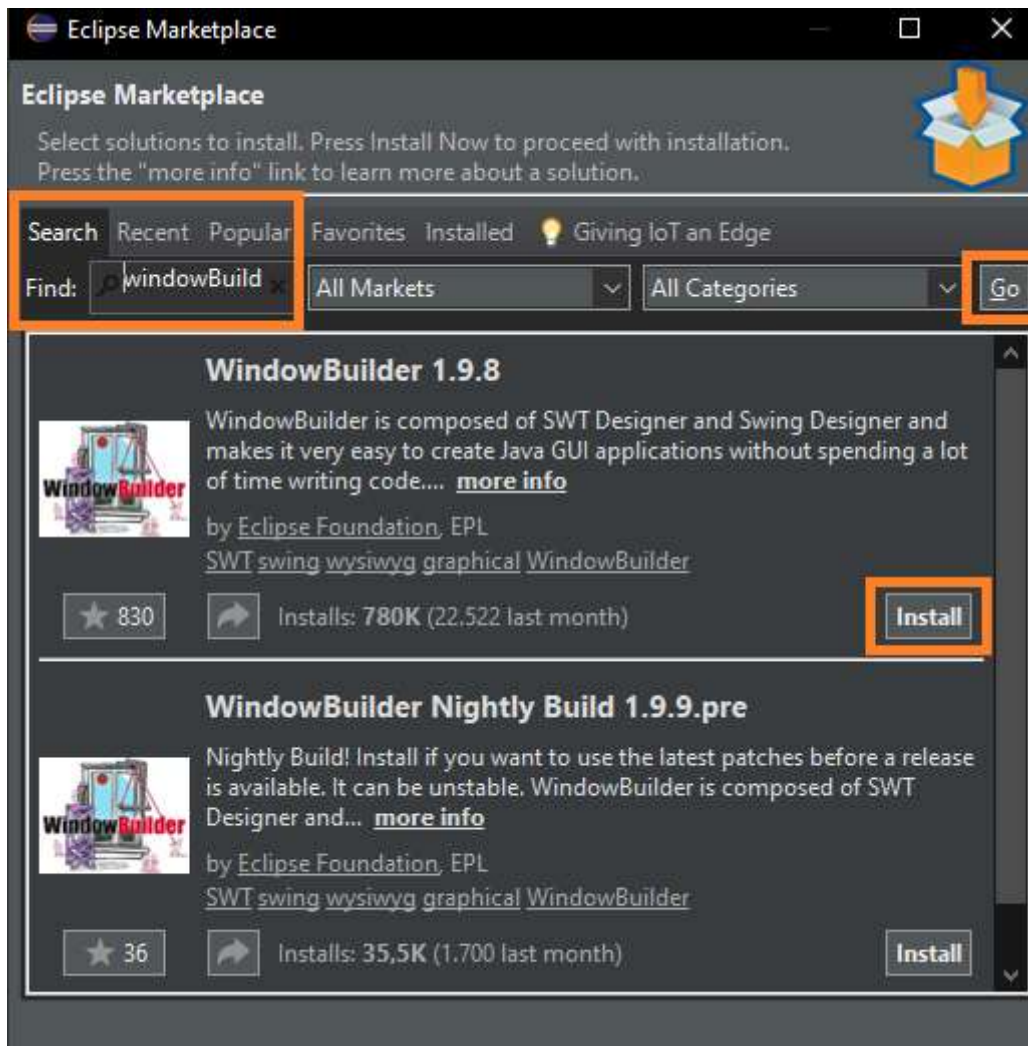
La versión 4.0 de **WindowBuilder** permitió que los componentes *SWT* y *Swing* se combinen usando las principales *APIs* de Java para construir ventanas.

## Instalación

La instalación es simple y rápida. El primer paso es acceder al menú **Help** dentro del editor y seleccionar **Eclipse Marketplace**:

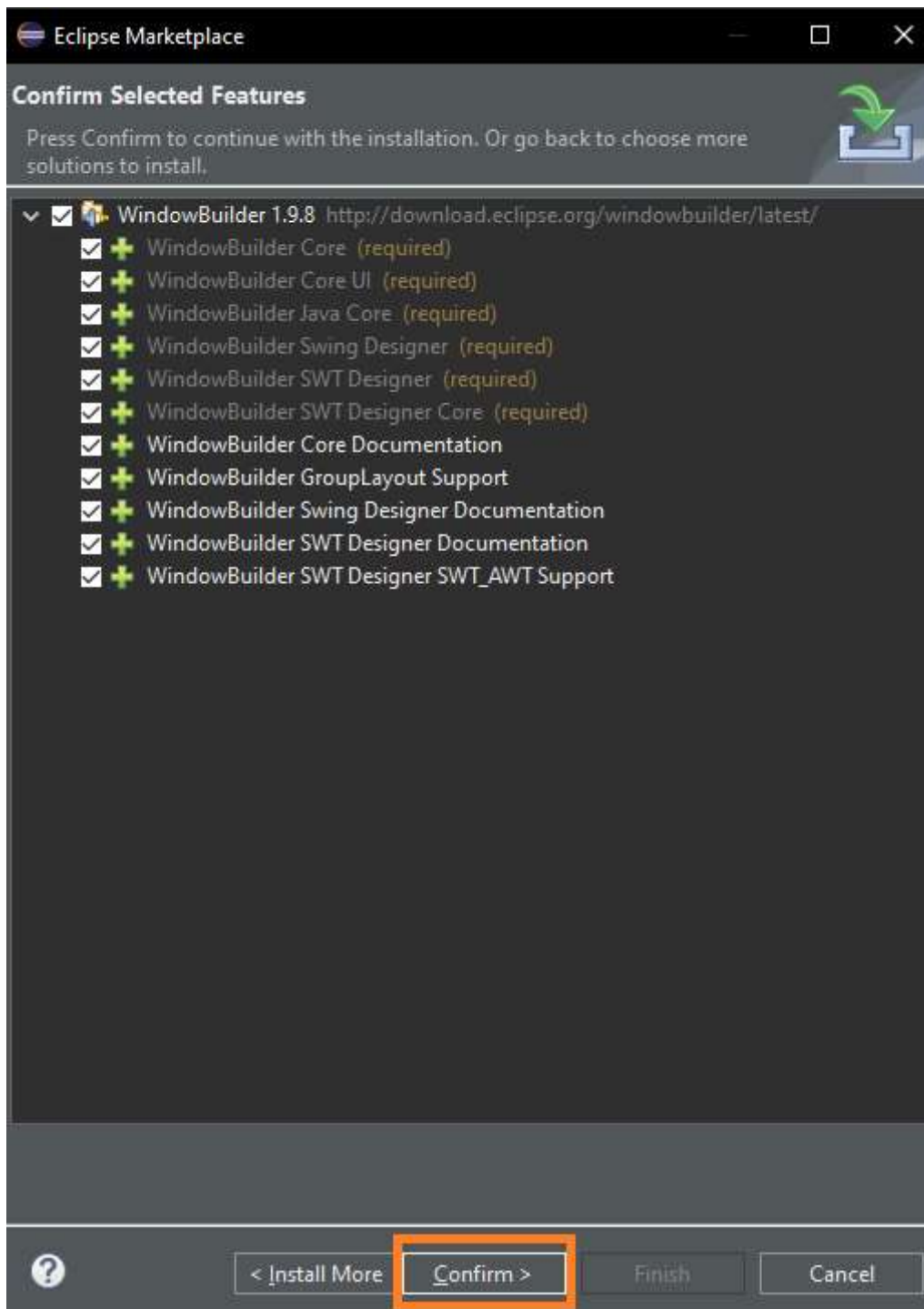


Luego se abrirá una ventana con un campo de búsqueda, simplemente escribe **windowbuilder**, haz clic en **Go** y se mostrará la última versión al principio. Finalmente, haz clic en **Install**:

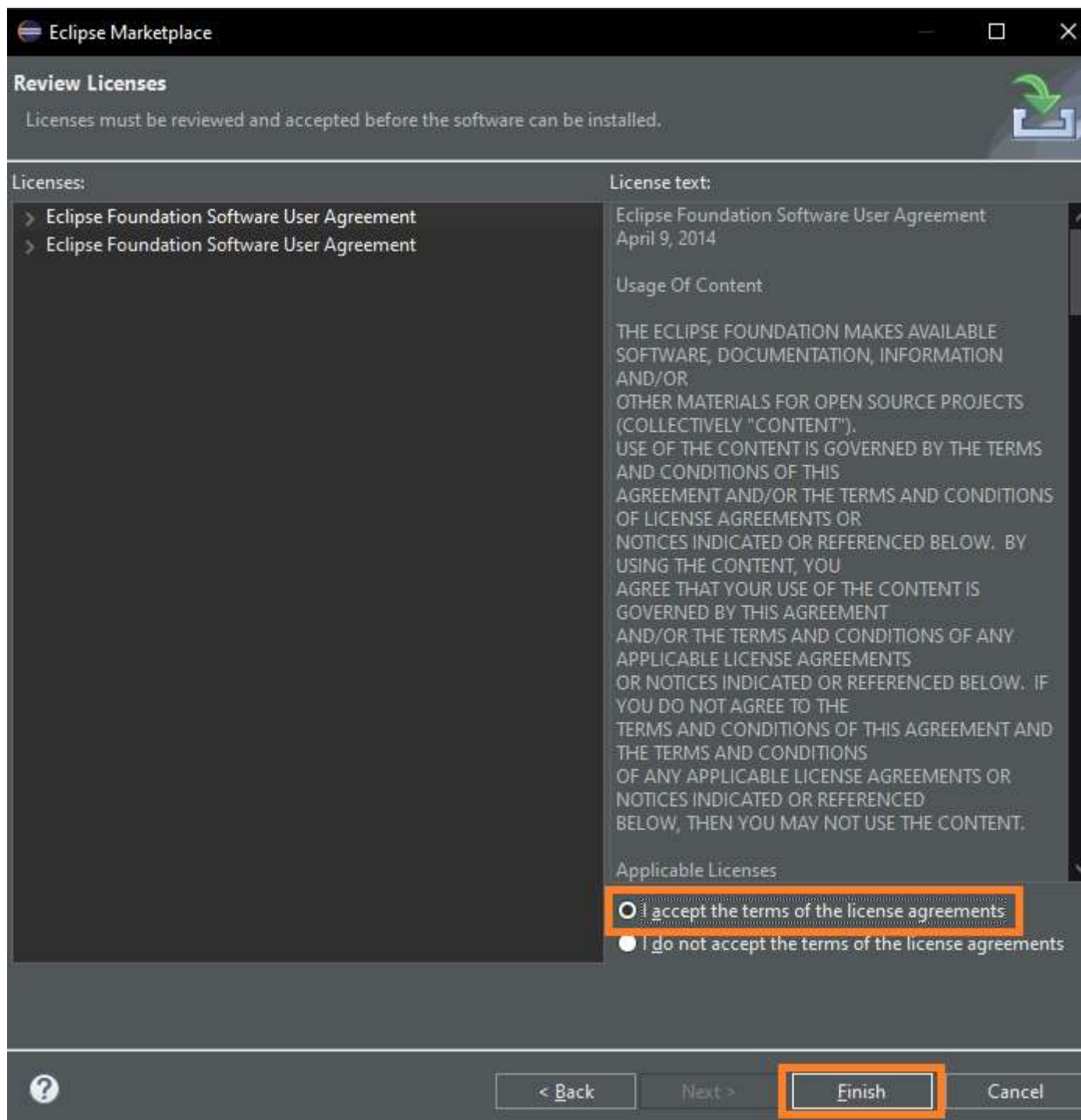


Selecciona todas las opciones y haz clic en **Confirm**:

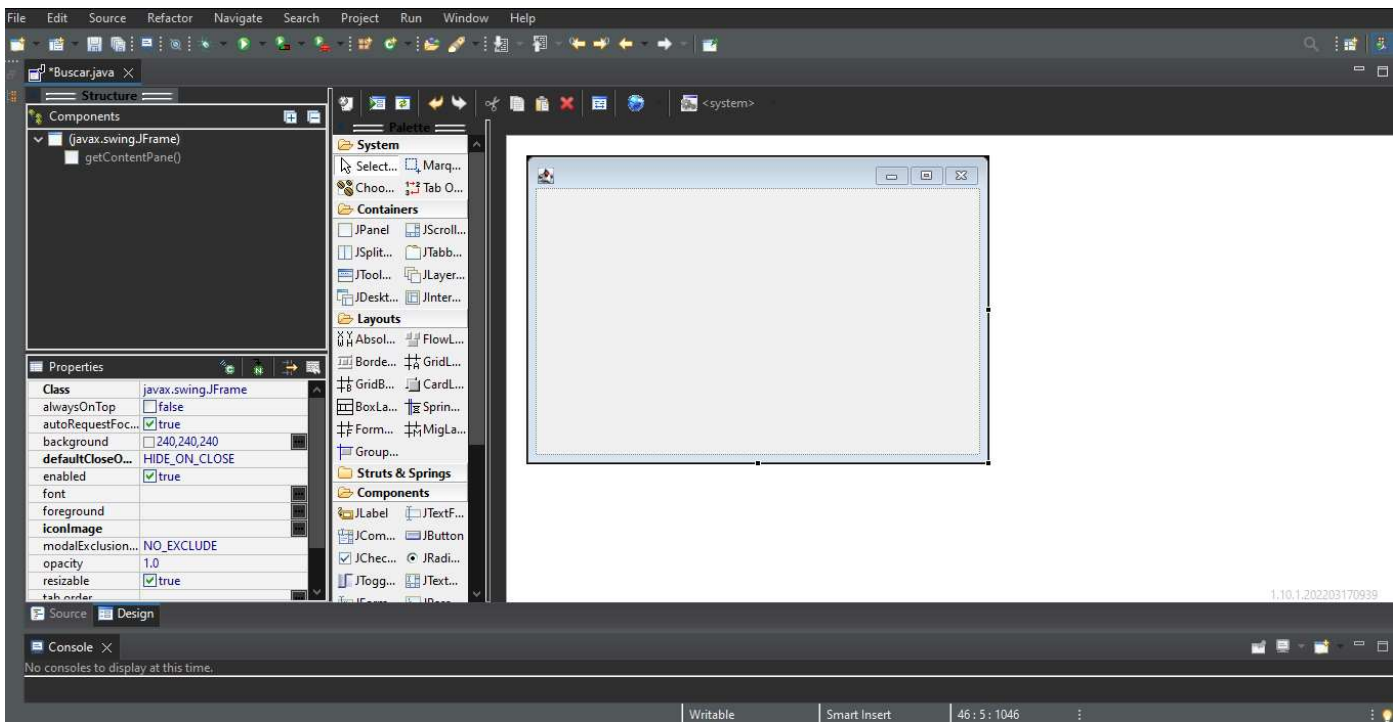




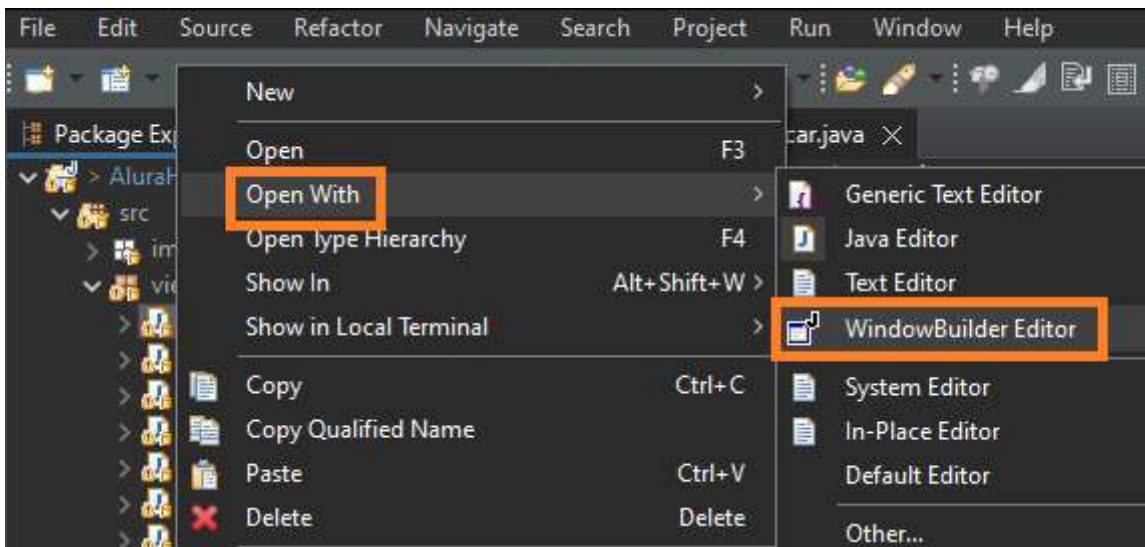
La siguiente ventana hará referencia a los términos de uso, siendo necesario aceptarlos para finalizar la instalación:



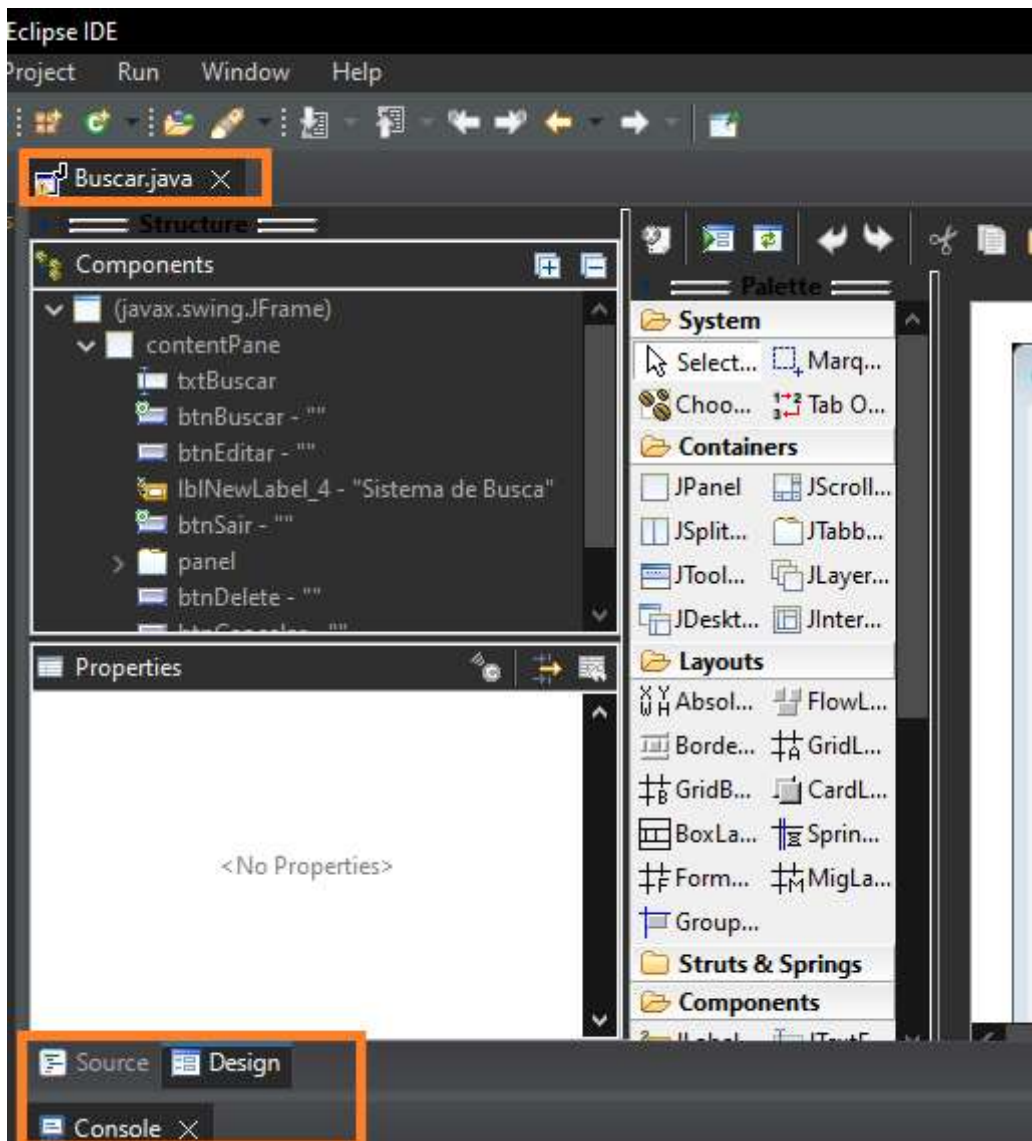
Después de completar el proceso, el editor se reiniciará. Esta será la presentación al usar el complemento:



Ahora bien, si descargas un proyecto listo realizado a través de WindowBuilder y deseas realizar cambios, puedes hacerlo directamente en el código, pero si lo prefieres a través de la interfaz gráfica, haz clic con el botón derecho del ratón sobre la clase con la que deseas trabajar y seleccione **OpenWith, WindowBuilder Editor**.



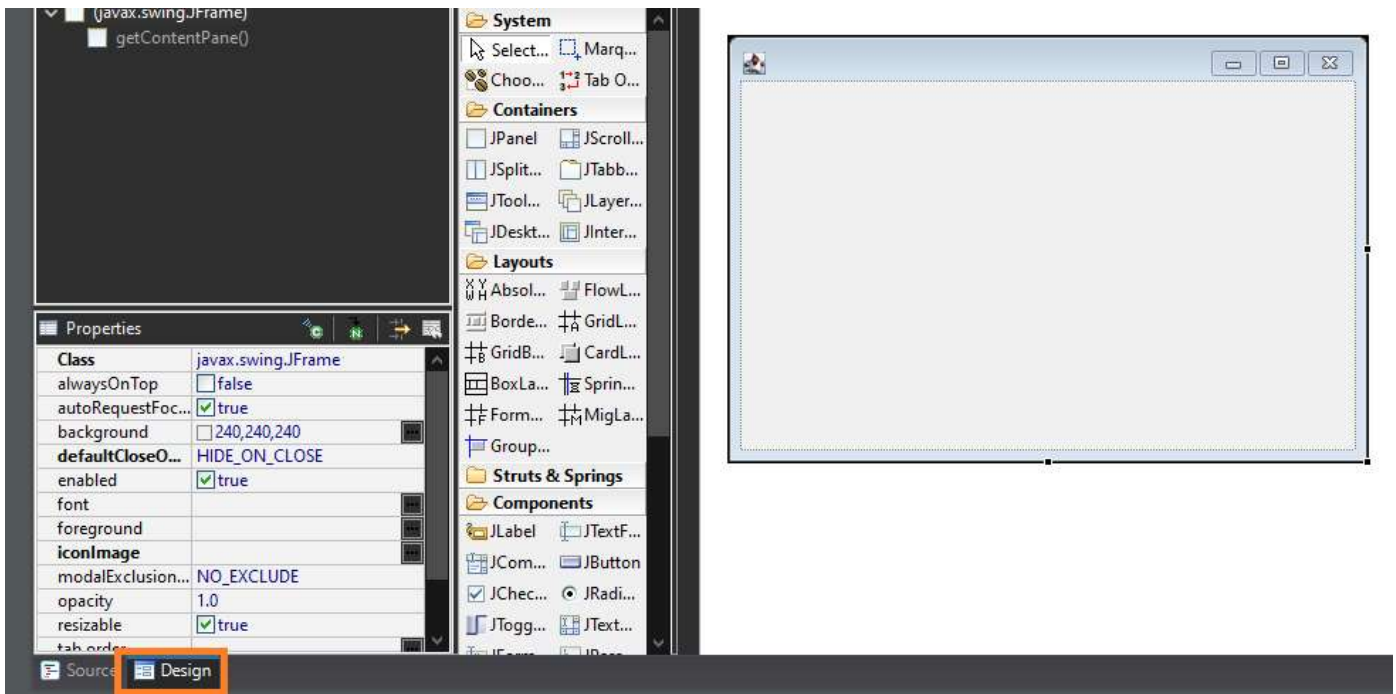
Esto abrirá la pestaña **Design** para manipulación, sobre el área de la consola.



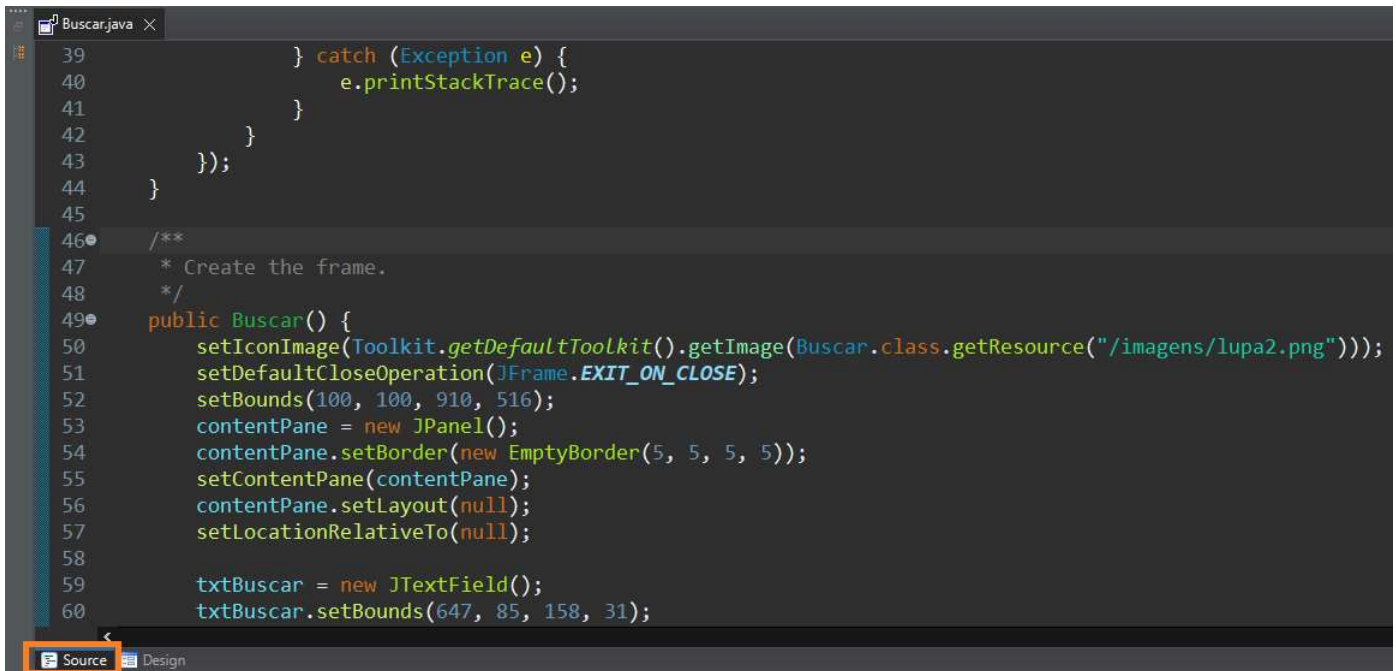
## Componentes

Se presentarán los principales componentes de su interfaz:

**Design View (Vista Previa del Diseño)** - es en este campo donde seguimos la construcción y el resultado de la interfaz gráfica:



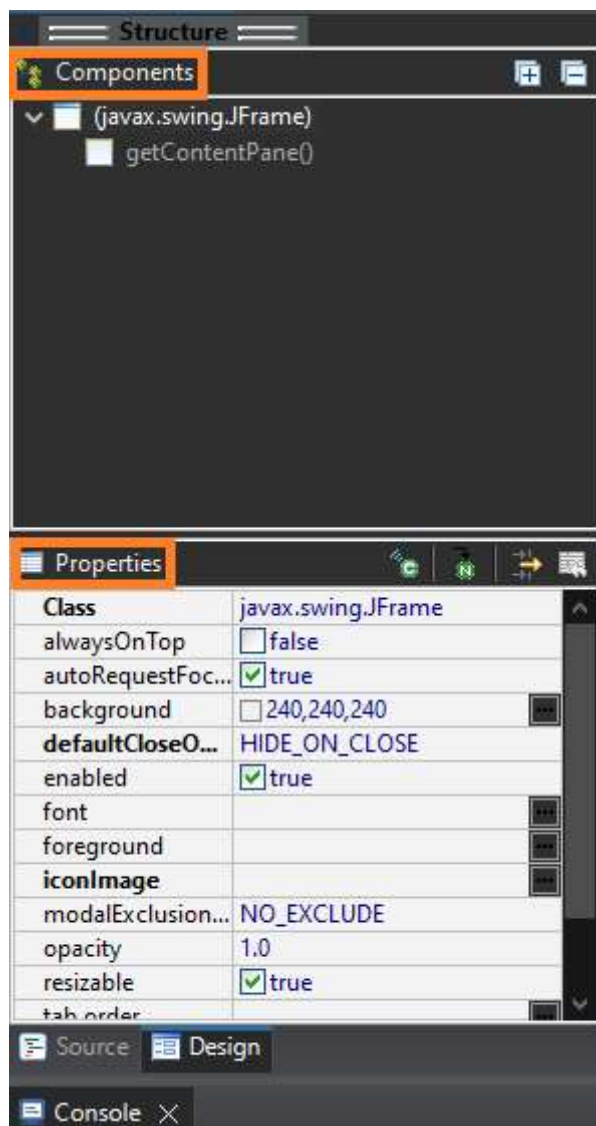
**Source View (Visualización del Diseño)** - espacio para escribir código y revisar lo que se generó automáticamente.



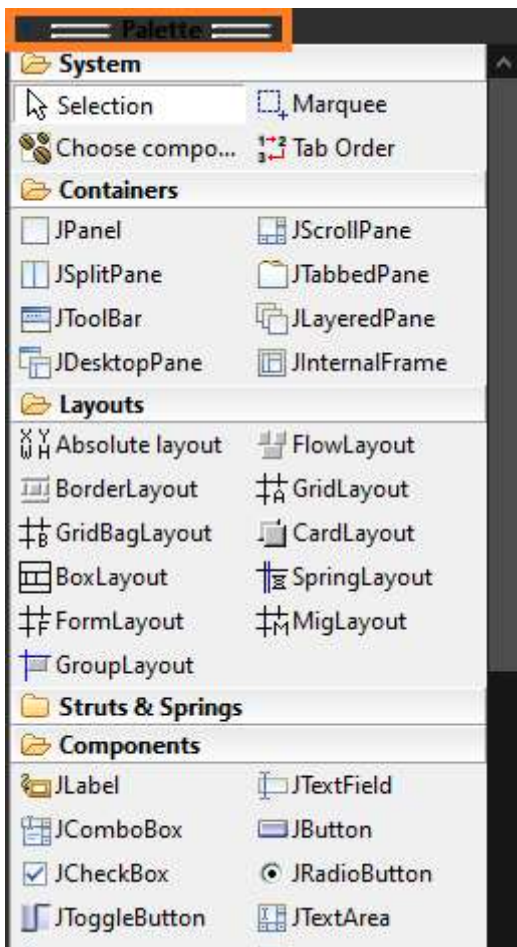
**Structure View (Vista de Estructura)** - consiste en el Árbol de Componentes y el Panel de Propiedades.

En el Árbol de componentes podemos ver la relación jerárquica entre ellos. Además de las propiedades, en el Panel de Propiedades se muestran los eventos de los componentes seleccionados.

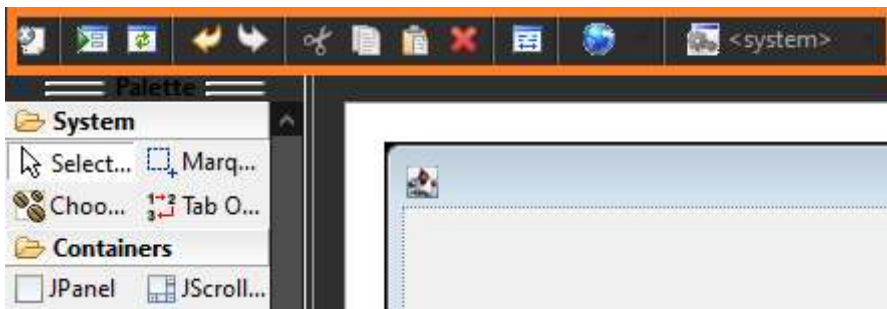




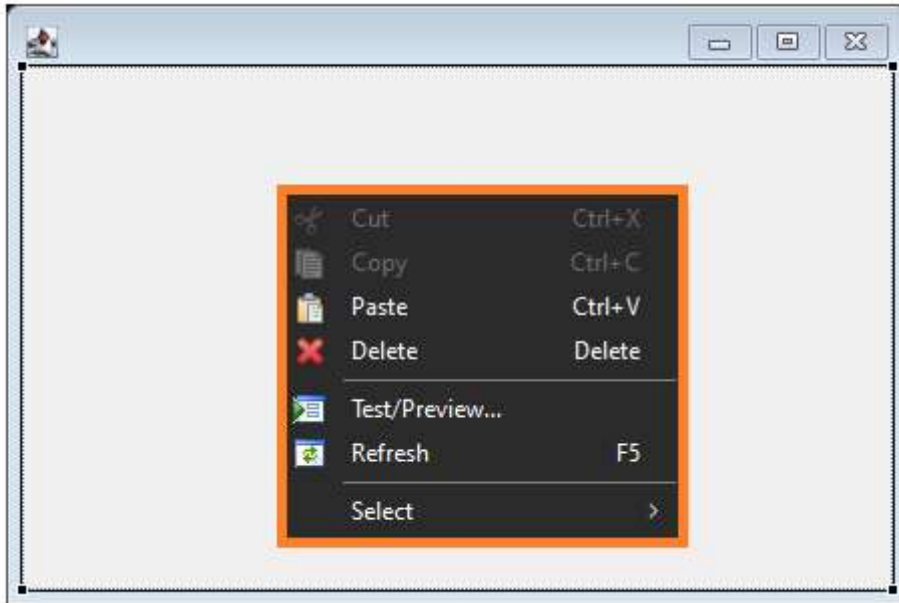
**Palette (Paleta)** - proporciona acceso a componentes específicos del *kit* de herramientas.



**Toolbar (Barra de Herramientas)** -\* proporciona acceso a los principales comandos utilizados.



**Context Menu (Menú Contextual)** - al igual que en la barra de herramientas, ofrece acceso a los principales comandos utilizados, pero con recursos más limitados.



## Look and Feel

Es un término en inglés, y el *look* que significa mirar, hace referencia al diseño de la aplicación. El *feel*, que significa sentir, está relacionado con la experiencia del usuario. El propósito de esta combinación es ayudar a referenciar la identificación de una marca y proponer un diseño intuitivo, con menos carga de memorización.

El *Look and Feel* predeterminado del editor es el *Metal*, con la misma presentación visual en todas las plataformas. Para usar uno diferente, es necesario configurar a través del método `UIManager.setLookAndFeel()` con el nombre especificado de la subclase apropiada de [Look and Feel](#) como argumento.

## Conclusión

Las posibilidades de crear proyectos usando Java son inmensas, además de ser un lenguaje poderoso, tiene una comunidad grande y comprometida y con varias herramientas de soporte.

Desarrollar con WindowBuilder no solo es simple, sino también divertido. Desde pequeñas ventanas de entrada hasta sistemas de gestión, él posibilita la investigación y práctica del lenguaje Java.



**Ellen Pimentel**

¡Hola, soy Ellen! Formo parte del equipo de Instructores Java de Alura Latam, me encanta aprender, y además del área de programación, me gusta estudiar sobre procesos y negocios.

## ARTÍCULOS DE TECNOLOGÍA

**En Alura encontrarás variados cursos sobre . ¡Comienza ahora!**

**SEMESTRAL****US\$49,90**

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana

- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 6 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

**ANUAL**

**US\$79,90**

un solo pago de US\$79,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas



Acceso a todo el contenido de la  
plataforma por 12 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

Acceso a todos  
los cursos

Estudia las 24 horas,  
dónde y cuándo quieras

Nuevos cursos  
cada semana

## NAVEGACIÓN

PLANES  
INSTRUCTORES  
BLOG  
POLÍTICA DE PRIVACIDAD  
TÉRMINOS DE USO  
SOBRE NOSOTROS  
PREGUNTAS FRECUENTES

## ¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

## BLOG

PROGRAMACIÓN  
FRONT END  
DATA SCIENCE  
INNOVACIÓN Y GESTIÓN  
DEVOPS

AOVS Sistemas de Informática S.A  
CNPJ 05.555.382/0001-33

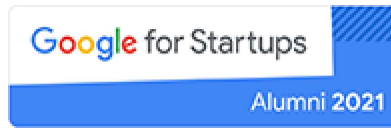
## SÍGUENOS EN NUESTRAS REDES SOCIALES



## ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

POWERED BY

## CURSOS

### Cursos de Programación

Lógica de Programación | Java

### Cursos de Front End

HTML y CSS | JavaScript | React

### Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

### Cursos de DevOps

Docker | Linux

### Cursos de Innovación y Gestión

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics | Liderazgo y Gestión de Equipos | Startups y Emprendimiento